Список литературы

1. Методы поиска ассоциативных правил [Электронный ресурс]. URL: http://www.intuit.ru/studies/courses/6/6/lecture/186 (дата обращения: 21.10.2017).
2. *Cheng P.*, *Lin C.-W.*, *Pan J.-S.* Use HypE to hide association rules by adding items. Shenzhen, 2015.
3. *Sathiyapriya K.*, *Sudha Sadasivam Dr. G.* A survey on privacy preserving association rule mining. Coimbatore, 2013.
4. *Sakenian Dehkordi M.*, *Naderi Dehkordi M.* Introducing an algorithm for use to hide sensitive association rules through perturb technique. Isfahan, 2016.

УДК 004.056

Р. В. Гибилинда
Научный руководитель: канд. тех. наук Д. А. Хорьков
Уральский федеральный университет, Екатеринбург

СПОСОБ ВОССТАНОВЛЕНИЯ ДАННЫХ В ФАЙЛОВОЙ СИСТЕМЕ EXT4 С ИСПОЛЬЗОВАНИЕМ ИНФОРМАЦИИ ЖУРНАЛА ИЗМЕНЕНИЙ ТОМА

Аннотация. В докладе рассмотрена проблема восстановления файлов в файловой системе ext4, указаны ее причины. Описан формат файла журнала изменений тома, позволяющего поддерживать раздел в исправном состоянии. Продемонстрирован способ, позволяющий как в ручном, так и в автоматическом режиме производить восстановление данных на разделе без использования сигнатурного поиска. Указаны достоинства и недостатки предложенного способа.

Ключевые слова: информация; восстановление; журналирование; доступность информации; ext4.

Основная сложность восстановления информации в ext4 связана с тем, что при удалении последней жесткой ссылки на файл драйвер заполняет нулями область индексного узла, где указаны номера кластеров с данными файл [1]. Анализ битовой карты с целью последующего исследования свободных областей памяти на машинном носителе занимает продолжительное время, особенно на накопителях большого объема и RAID-массивах. Для ускорения процесса поиска данных удаленного файлового объекта требуется возможность быстрого обнаружения номеров его кластеров. Ее предоставляет журнал изменений тома — файл, призванный обеспечить отказоустойчивость раздела с точки зрения его логической структуры. Формат журнала и алгоритм его работы не

являются закрытыми и описываются на сайте Ext4 Wiki [1]. Алгоритм восстановления информации в комбинации с утилитой для сигнатурного поиска файлов foremost описан в работе [2], однако подобного материала в отношении ext4 не представлено.

Файл журнала (рис. 1) записывается циклично, а информация расположена в блоках различного типа, в результате чего обеспечивается непрерывный контроль над состоянием производимых файловых операций. В связи с тем, что при каждом монтировании раздела для записи данные в журнал начинают записываться с его начала, подключать раздел с целью извлечения содержащейся на нем информации необходимо только в режиме чтения [3, с. 389].

При совершении файловых операций содержимое кластера, где будут располагаться изменения, и информация об этих изменениях сначала попадают в журнал, затем измененный кластер записывается на диск, а в журнале создается *блок подтверждения*, означающий, что операция прошла успешно. При сбое записи создается *блок отзыва* операций с номерами кластеров, содержимое которых необходимо повторно записать на диск при загрузке операционной системы во избежание повреждения логической целостности раздела.

Для большей наглядности рассмотрим пример расположения информации в журнале и поиска состояний индексного узла (inode) удаленного файла на рис. 2.

Для восстановления данных удаленных файлов необходимо использовать кластеры, помещенные в журнал. Их номера можно определить в массиве описателей. Поскольку находящиеся в журнале копии содержимого кластеров могут включать в себя фрагменты таблиц индексных узлов, они представляют особый интерес в контексте восстановления информации. Наибольшее зна-

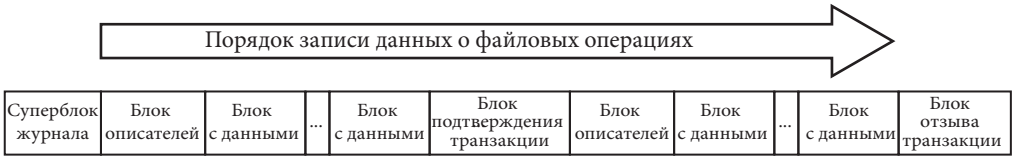

Рис. 1. Формат журнала файловой системы ext4 [1, 2]

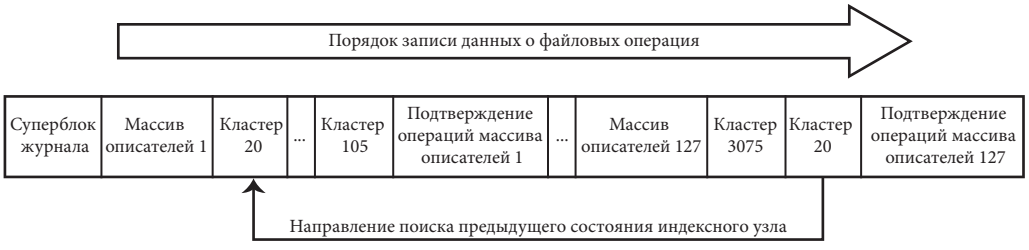

Рис. 2. Пример расположения данных о файловых операциях

чение в этом случае имеют узлы удаленных файлов, по причине возможного наличия в журнале кластеров с данными, которые файлы занимали.

При использовании предложенного способа сначала нужно извлечь с помощью утилиты dd журнал. Его inode, по умолчанию имеющий номер 8, следует использовать для определения расположения кластеров журнала утилитой debugfs.

Вторым этапом получения содержимого удаленного файла является поиск в журнале его индексного узла, который, например, расположен в кластере 20 после массива описателей 127 (рис. 2).

Далее нужно осуществить поиск и извлечение номеров блоков с данными файла. В связи с тем, что состояние найденного на 2 этапе inode соответствует удаленному файлу, поля с номерами кластеров заполнены нулями [4]. Для того чтобы их обнаружить, необходимо найти этот же узел в ранее описанных в журнале операциях, когда файл еще не был удален, например, в том же кластере с номером 20 после массива описателей 1 (рис. 2). Наличие в журнале нескольких версий двадцатого кластера обусловлено тем, что, помимо индексного узла удаленного файла, в нем могут присутствовать до 15 аналогичных структур, связанных с другими файловыми объектами. Вследствие того, что при изменении любого inode, который расположен в одном кластере с найденным на 2-м этапе индексным узлом, происходит копирование этого блока с данными в журнал, необходимо учитывать, что вероятность нахождения информации, описывающей состояние файла до его удаления, зависит от наличия изменений соседних узлов.

Если информация о предыдущем состоянии inode будет обнаружена в журнале, содержимое кластеров, на которые он указывает, следует скопировать в новый файл с помощью утилиты dd.

Эффективность предложенного способа значительно зависит от интенсивности файловых операций на разделе, при этом наилучшие результаты достигаются при быстром реагировании на удаление важной информации. В ходе проведенного эксперимента установлено, что при удалении файла на рабочей станции запись об этом событии находилась в журнале в течение трех часов, в то же время небольшой Web-сервер, обслуживающий форум, «переписывает» журнал менее чем за 20 минут. К недостаткам описанного способа восстановления можно отнести отсутствие реального имени файлового объекта у извлеченных данных, а также наличие ложных срабатываний при поиске удаленных индексных узлов в содержимом журнала. Среди достоинств способа стоит отметить возможность его полной автоматизации с помощью, например скриптового файла, и необходимость использования всего двух дополнительных утилит (dd и debugfs), которые входят в стандартную поставку большинства дистрибутивов операционных систем на базе ядра Linux.

Список литературы

1. Ext 4 Disk Layout [Электронный ресурс] // Ext4 (and Ext2/Ext3) Wiki. URL: https://ext4.wiki.kernel.org/index.php/Ext4_Disk_Layout (дата обращения: 28.10.2017).

2. *Narvaez G.* Taking advantage of Ext3 journaling file system in a forensic investigation // SANS Institute, 2007. URL: https://www.sans.org/reading-room/whitepapers/forensics/advantage-ext3-journaling-file-system-forensic-investigation-2011 (дата обращения: 10.10.2017).

3. *Кэрриэ Б.* Криминалистический анализ файловых систем. СПб. : Питер, 2007. 480 с.

4. *Бакланов В. В.* Защитные механизмы операционной системы Linux : учеб. пособие / под ред. Н. А. Гайдамакина. Екатеринбург : УрФУ, 2011. 354 с.